Supervised Learning

Filter Methods

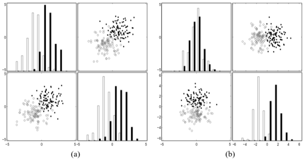
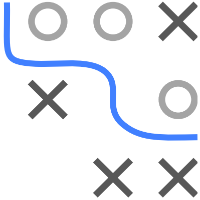


Figure 1: **Information gain from presumably redundant variables.** (a) A two class problem with independently and identically distributed (i.i.d.) variables. Each class has a Gaussian distribution with no covariance. (b) The same example after a 45 degree rotation showing that a combination of the two variables yields a separation improvement by a factor $\sqrt{2}$. I.i.d. variables are not truly redundant.

Learning goals

- Understand how filter methods work
- Understand how to apply them for feature selection
- Understand advantages and disadvantages, and how to overcome them.

INTRODUCTION



- **Filter methods** construct a measure that describes the dependency between a feature and the target variable.
- They Yield a numeric score for each feature j, known as **variable-ranking**.
- They are model-agnostic and can be applied generically.
- Filter methods are strongly related to methods for determining variable importance.

χ^2-STATISTIC

- Test for independence between the j-th feature and the target y.
- Numeric features or targets need to be discretized.
- Hypotheses:

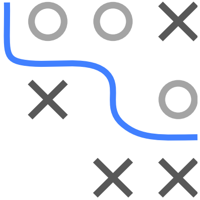
  $H_0 : p(x_j = l, y = k) = p(x_j = l)\, p(y = k), \forall j = 1, \dots, k_1$
  $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall k = 1, \dots, k_2$

  $H_1 : \exists\, j, k : p(x_j = l, y = k) \neq p(x_j = l)\, p(y = k)$
- Calculate the χ^2-statistic for each feature-target combination:

$$\chi^2 = \sum_{j=1}^{k_1} \sum_{k=1}^{k_2} \left(\frac{e_{jk} - \tilde{e}_{jk}}{\tilde{e}_{jk}}\right) \quad \underset{approx.}{\overset{H_0}{\sim}} \quad \chi^2((k_1 - 1)(k_2 - 1))$$

  where e_{jk} is the observed relative frequency of pair (j, k) and $\tilde{e}_{jk} = \frac{e_{j.}\, e_{.k}}{n}$ is the expected relative frequency.
- The greater χ^2, the more dependent is the feature-target combination, the more relevant is the feature.

PEARSON & SPEARMAN CORRELATION



Pearson correlation $r(\mathbf{x}_j, y)$:

- For numeric features and targets only.
- Most sensitive for linear or monotonic relationships.
- $r(\mathbf{x}_j, y) = \frac{\sum_{i=1}^{n}(x_j^{(i)} - \bar{x}_j)(y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_j^{(i)} - \bar{x}_j)}\sqrt{(\sum_{i=1}^{n} y^{(i)} - \bar{y})}}, \qquad -1 \leq r \leq 1$

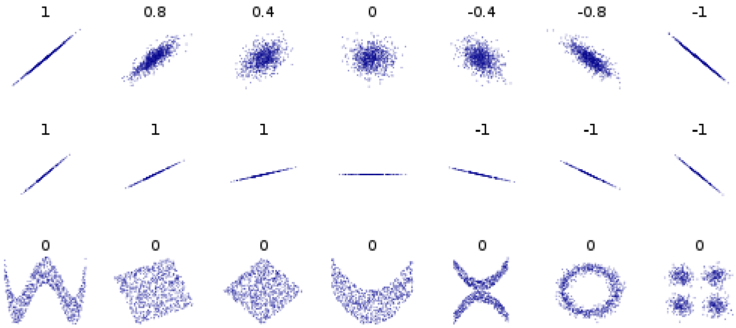
Spearman correlation $r_{SP}(\mathbf{x}_j, y)$:

- For features and targets at least on an ordinal scale.
- Equivalent to Pearson correlation computed on the ranks.
- Assesses monotonicity of the dependency relationship.

Use absolute values $|r(\mathbf{x}_j, y)|$ for feature ranking: higher score indicates a higher relevance.

PEARSON & SPEARMAN CORRELATION

Only **linear** dependency structure, non-linear (non-monotonic) aspects are not captured:



DISTANCE CORRELATION

$$r_D(\mathbf{x}_j, y) = \sqrt{\frac{c_D^2(\mathbf{x}_j, y)}{\sqrt{c_D^2(\mathbf{x}_j, \mathbf{x}_j)c_D^2(y, y)}}}:$$

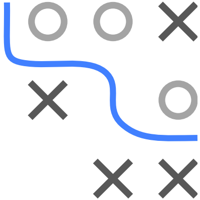
Normed version of **distance covariance**

$$c_D(\mathbf{x}_j, y) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{k=1}^{n} D_{\mathbf{x}_j}^{(ik)} D_y^{(ik)}$$

$$D_x^{(ik)} = d\left(x_j^{(i)}, x_j^{(k)}\right) - (\bar{d}_{xj}^{(i\cdot)} + \bar{d}_{xj}^{(\cdot k)} - \bar{d}_{xj}^{(\cdot\cdot)})$$

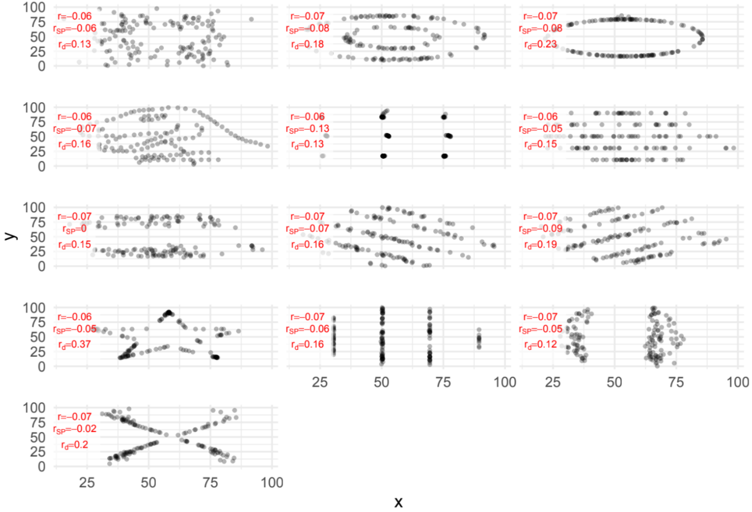
- $D_x^{(ik)}$ are the centered pairwise distances.
- $d\left(x_j^{(i)}, x_j^{(k)}\right)$ represent the distances of observations.
- $\bar{d}_{xj}^{(i\cdot)} = \frac{1}{n}\sum_{k=1}^{n} d\left(x_j^{(i)}, x_j^{(k)}\right)$ represent the mean distances.

DISTANCE CORRELATION

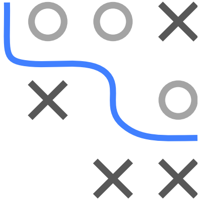


- $0 \leq r_D(\mathbf{x}_j, y) \leq 1 \quad \forall j \in \{1, \ldots, p\}$
- $r_D(\mathbf{x}_j, y) = 0$ only if $\mathbf{x}$ and y are empirically independent (!)
- $r_D(\mathbf{x}_j, y) = 1$ for exact linear dependencies
- Assesses strength of **non-monotonic**, **non-linear** dependencies
- Gennerally applicable, even for ranking multivariate features or non-tabular inputs (text, images, audio, etc.)
- Expensive to compute for large data.

DISTANCE CORRELATION



WELCH'S t-TEST

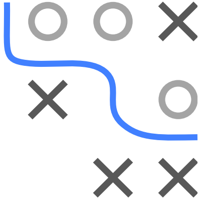


- For binary classification with numeric features.
- Test for unequal means of the j-th feature.
- Let $\mathcal{Y} \in \{0, 1\}$. The subscript j_0 refers to the j-th feature where $y = 0$ and j_1 where $y = 1$.
- Hypotheses:
  H_0: $\mu_{j_0} = \mu_{j_1}$ vs. H_1: $\mu_{j_0} \neq \mu_{j_1}$
- Calculate Welch's t-statistic for every feature $\mathbf{x}_j$

$$t_j = \frac{\bar{x}_{j_0} - \bar{x}_{j_1}}{\sqrt{\left(\frac{S^2_{x_{j_0}}}{n_0} + \frac{S^2_{x_{j_1}}}{n_1}\right)}}$$

  where $\bar{x}_{j_0}$, $S^2_{x_{j_0}}$ and n_0 are the sample mean, the population variance and the sample size for y = 0, respectively.
- A higher t-score indicates higher relevance of the feature.

F-TEST



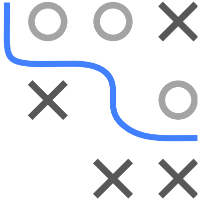
- For multiclass classification ($g \geq 2$) and numeric features.
- Assesses whether the expected values of a feature $\mathbf{x}_j$ within the classes of the target differ from each other.
- Hypotheses:
  $H_0 : \mu_{j_0} = \mu_{j_1} = \cdots = \mu_{j_g}$ vs. $H_1 : \exists\, k, l : \mu_{j_k} \neq \mu_{j_l}$
- Calculate the F-statistic for each feature-target combination:

$$F = \frac{\text{between-group variability}}{\text{within-group variability}}$$

$$F = \frac{\sum_{k=1}^{g} n_k (\bar{x}_{j_k} - \bar{x}_j)^2 / (g-1)}{\sum_{k=1}^{g} \sum_{i=1}^{n_k} (x_{j_k}^{(i)} - \bar{x}_{j_k})^2 / (n-g)}$$

  where $\bar{x}_{j_k}$ is the sample mean of feature $\mathbf{x}_j$ where $y = k$ and $\bar{x}_j$ is the overall sample mean of feature $\mathbf{x}_j$.
- A higher F-score indicates higher relevance of the feature.

MUTUAL INFORMATION

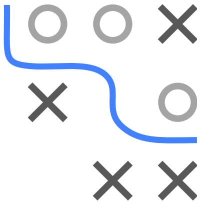


$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right]$$

- Each variable $\mathbf{x}_j$ is rated according to $I(\mathbf{x}_j; y)$, this is sometimes called information gain.
- MI is a measure of the amount of "dependence" between variables. It is zero if and only if the variables are independent.
- On the other hand, if one of the variables is a deterministic function of the other, the mutual information is maximal.
- Not limited to real-valued random variables.
- More general measure of dependence between variables than correlation.

USING FILTER METHODS

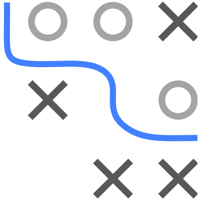
1. Calculate filter-values.
2. Sort features by value.
3. Train model on $\tilde{p}$ best features.



How to choose $\tilde{p}$?

- It can be prescribed by the application.
- Eyeball estimation: Read from filter plots (i.e., Scree plots).
- Use resampling.

USING FILTER METHODS



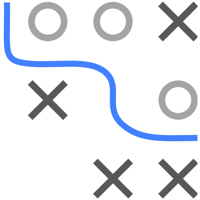
Advantages:

- Easy to calculate.
- Typically scales well with the number of features p.
- Generally Interpretable.
- Model-agnostic.

Disadvantages:

- Univariate analysis may ignore multivariate dependencies.
- Redundant features will have similar weights.
- Ignores the learning algorithm.

MINIMUM REDUNDANCY MAXIMUM RELEVANCY



- Most filter type methods features based on a certain filter method without considering relationships among the features.
  - Features may be correlated and hence, may cause redundancy.
  - Selected features cover narrow regions in space.
- We want the features to be relevant and maximally dissimilar to each other (minimum redundancy).
- Features can be either continuous or categorical.

mRMR: CRITERION FUNCTIONS

- Let $S \subset \{1, \ldots, p\}$ be a subset of features we want to find.

$$\min \operatorname{Red}(S), \quad \operatorname{Red}(S) = \frac{1}{|S|^2} \sum_{j,l \in S} I_{xx}(\mathbf{x}_j, \mathbf{x}_l)$$

$$\max \operatorname{Rel}(S), \quad \operatorname{Rel}(S) = \frac{1}{|S|} \sum_{j \in S} I_{xy}\left(\mathbf{x}_j, \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\right\}\right)$$

- I_{xx} measures the strength of the dependency between two features.
- I_{xy} measures the strength of the dependency between a feature and the target.
- They could be mutual information, correlation, F-statistic, etc.

mRMR: CRITERION FUNCTIONS

- To optimize simultainously, the criteria is combined into a single objective function:

$$\Psi(S) = (\mathrm{Rel}(S) - \mathrm{Red}(S)) \quad \text{or} \quad \Psi(S) = (\mathrm{Rel}(S)/\mathrm{Red}(S))$$

- Exact solution requires $\mathcal{O}(|\mathcal{X}|^{|S|})$ searches, where $|\mathcal{X}|$ is the number of features and $|S|$ is the number of selected features.

In practice, incremental search methods are used to find near-optimal feature sets defined by Ψ:

- Suppose we already have a feature set with $m-1$ features S_{m-1}.
- Next, we select the m-th feature from the set $\bar{S}_{m-1}$ by selecting the feature that maximizes:

$$\max_{j \in \bar{S}_{m-1}} [I_{xy}(\mathbf{x}_j, \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\right\}) - \frac{1}{|S_{m-1}|} \sum_{l \in S_{m-1}} I_{xx}(\mathbf{x}_j, \mathbf{x}_l)]$$

- The complexity of this incremental algorithm is $\mathcal{O}(|p| \cdot |S|)$.

mRMR: ALGORITHM

Algorithm mRMR algorithm

Set $S = \emptyset$, $R = \{1, \dots, p\}$

Find the feature with maximum relevancy:

$$j^* := \arg\max_j I_{xy}(\mathbf{x}_j, \left\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}\right\})$$

Set $S = \{j^*\}$ and update $R \leftarrow R \setminus \{j^*\}$

repeat

  Find feature $\mathbf{x}_j$ that maximizes:

$$\max_{j \in R} [I_{xy}(\mathbf{x}_j, \left\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}\right\}) - \frac{1}{|S|} \sum_{l \in S} I_{xx}(\mathbf{x}_j, \mathbf{x}_l)]$$

  Update $S \leftarrow S \cup \{j^*\}$ and $R \leftarrow R \setminus \{j^*\}$

until Expected number of features have been obtained or some other constraints are satisfied.